METHOD AND SYSTEM FOR NON-INTRUSIVE CODE COVERAGE

TECHNICAL FIELD

[0001] Embodiments of the invention relate to software testing, and more specifically to non-intrusive code coverage.

BACKGROUND

[0002] Code coverage tools are used to collect information about software testing to determine whether a software program has been tested thoroughly. Typically, code coverage tools instrument code into a target program in order to collect code coverage information. The code coverage code is either instrumented into the target program source code or into the target executable directly. This instrumentation is intrusive to the target program and increases the size of the target program.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

FIG. 1 is a block diagram illustrating a suitable computing environment in which certain aspects of the illustrated invention may be practiced.

FIG. 2 is a block diagram illustrating one generalized embodiment of a system incorporating the invention.

FIG. 3 is a flow diagram illustrating a method according to an embodiment of the invention.

DETAILED DESCRIPTION

[0003] Embodiments of a system and method for non-intrusive code coverage are described. In the following description, numerous specific details are set forth. However, it is understood that embodiments of the invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to obscure the understanding of this description.

[0004] Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. Thus, the appearances of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0005] Fig. 1 is a block diagram illustrating a suitable computing environment in which certain aspects of the illustrated invention may be practiced. Methods of the invention may be implemented on a computer system 100 having components 102 – 112, including a central processing unit (CPU) 102, a memory 106, an Input/Output device 104, a data storage device 112, and a network interface 110, coupled to each other via a bus 108. The components perform their conventional functions known in the art and provide the means for implementing the system 100. Collectively, these components

represent a broad category of hardware systems, including but not limited to general purpose computer systems, mobile or wireless computing systems, and specialized packet forwarding devices. It is to be appreciated that various components of computer system 100 may be rearranged, and that certain implementations of the present invention may not require nor include all of the above components. Furthermore, additional components may be included in system 100, such as additional processors, storage devices, memories (e.g. RAM, ROM, or flash memory), and network or communication interfaces.

[0006] As will be appreciated by those skilled in the art, the content for implementing an embodiment of the method of the invention, for example, computer program instructions, may be provided by any machine-readable media which can store data that is accessible by system 100, as part of or in addition to memory, including but not limited to cartridges, magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read-only memories (ROMs), and the like. In this regard, the system 100 is equipped to communicate with such machine-readable media in a manner well-known in the art.

[0007] It will be further appreciated by those skilled in the art that the content for implementing an embodiment of the method of the invention may be provided to the system 100 from any external device capable of storing the content and communicating the content to the system 100. For example, in one embodiment of the invention, the system 100 may be connected to a network, and the content may be stored on any device in the network.

[0008] Fig. 2 illustrates a system incorporating an embodiment of the invention. System 200 includes a virtual machine (VM) 202. The virtual machine 202 may be any

platform or operating system (OS) and provides a virtualization environment to run target programs, such as 204. System 200 also includes a virtual machine monitor (VMM) 206. The virtual machine monitor 206 maintains information about the virtual machine 202 and monitors programs running on the virtual machine. The virtual machine monitor 206 includes an information collection module (ICM) 208 and a buffer 210. The information collection module 208 configures the CPU 102 to notify the information collection module 208 when a branch is taken. In one embodiment, a configuration register is set to provide a notification to the information collection module 208 when a branch is taken. In one embodiment, the CPU 102 notifies the information collection module via an interruption. When the information collection module 208 is notified by the CPU 102 about the branch, the information collection module 208 records the branch address and stores the information in the buffer 210. After runtime, the execution information 218 collected and stored in the buffer 210 may be input into a coverage pattern generation module (CPGM) 212 along with the target program source file or symbol file 220. The CPGM 212 may be any graphical user interface (GUI) that converts the target program source file 220 and the execution information 218 into code coverage statistics that may be displayed to a user.

[0009] Fig. 3 illustrates a method according to one embodiment of the invention. At 300, a CPU is configured to notify an information collection module when a branch is taken. At 302, a branch address is recorded when the information collection module is notified of a branch taken. At 304, the recorded branch addresses are stored to determine code coverage of a target program on the CPU. In one embodiment, the recorded branch addresses are stored in a buffer. Then, after runtime, the recorded branch addresses may

be sent to a CPGM along with the target program source file to display code coverage statistics. By recording and analyzing the branches taken, the source code of the target program that was executed may be determined. This information may then aid in determining whether software testing was adequate in covering the code of the target program.

[0010] While the invention has been described in terms of several embodiments, those of ordinary skill in the art will recognize that the invention is not limited to the embodiments described, but can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting.
